CLAIMS

What is claimed is:

1. An apparatus comprising:

a mask generator to generate a mask field for an operand having a word length, the mask field defining an operand field within the operand to be operated by an operation, the operand field having a field length; and

an execution unit coupled to the mask generator to execute the operation on the operand field.

2. The apparatus of claim 1 wherein the mask generator comprises:

a first decoder to decode a begin position specifier into a begin bit pattern;

a second decoder to decode an end position specifier into an end bit pattern; and

a logic circuit coupled to the first and second decoders to combine the begin and end bit patterns into the mask field having the field length limited by the begin and end positions.

3. The apparatus of claim 1 wherein the execution unit comprises:

a field arithmetic logic unit (ALU) to generate an ALU result using one of an arithmetic and logic operations on the operand field of at least one of first and second ALU operands.

4. The apparatus of claim 3 wherein the execution unit further comprises:

an operand selector to select a selector operand from a source operand and an immediate operand, the source operand being from a register file.

5. The apparatus of claim 4 wherein the execution unit further comprises:

a first barrel shifter coupled to the operand selector to shift the selector operand to generate the first ALU operands according to one of the begin and end positions.

6. The apparatus of claim 5 wherein the execution unit further comprises:

a second barrel shifter coupled to the field ALU to shift the ALU result.

7. The apparatus of claim 5 wherein the execution unit further comprises:
a context selector coupled to the field ALU to select a field result, on a bit-by-bit basis according to the operand field, from at least one of the first and second ALU operands and the ALU result.

8. The apparatus of claim 6 wherein the execution unit further comprises:
a context selector coupled to the field ALU to select a field result, on a bit-by-bit basis according to the operand field, from at least one of the first and second ALU operands, the ALU result, and the shifted ALU result.

9. The apparatus of claim 3 wherein the field ALU comprises:
N single bit ALUs connected in cascade to generate the field result, the field result including a single bit ALU result.

10. The apparatus of claim 9 wherein the field result includes at least a condition code representing a condition of the field result.

11. The apparatus of claim 9 wherein the single bit ALU comprises:
an adder/ subtractor to perform an add/ subtraction on the first and second ALU operands and generate a carry output.

12. The apparatus of claim 11 wherein the single bit ALU further comprising:
a zero section to generate a zero condition code using a carry from a less significant section;
a negative section to generate a out sign bit for the field result using current and next operand fields; and
an overflow section to generate an overflow bit for the field result using the next operand field.

13. The apparatus of claim 2 further comprising:
a field specifier selector coupled to the mask generator to generate at least one of the begin and end position specifiers.

14. The apparatus of claim 13 wherein the field specifier selector generates the at least one of the begin and end position specifiers from at least one of an instruction specifying the operation, a general-purpose register, a special-purpose register, and a configuration register.

15. The apparatus of claim 1 wherein the operand field is one of a contiguous field and a non-contiguous field.

16. A method comprising:

generating a mask field for an operand having a word length, the mask field defining an operand field within the operand to be operated by an operation, the operand field having a field length; and

executing the operation on the operand field, leaving portion outside the operand field unchanged.

17. The method of claim 16 wherein generating the mask field comprises:

decoding a begin position specifier into a begin bit pattern;

decoding an end position specifier into an end bit pattern; and

combining the begin and end bit patterns into the mask field having the field length limited by the begin and end positions.

18. The method of claim 16 wherein executing the operation comprises:

generating an ALU result using one of an arithmetic and logic operations on the operand field of at least one of first and second ALU operands.

19. The method of claim 18 wherein executing the operation further comprises:

selecting a selector operand from a source operand and an immediate operand, the source operand being from a register file.

20. The method of claim 19 wherein executing the operation further comprises:

shifting the selector operand to generate the first ALU operands according to one of the begin and end positions.

21. The method of claim 20 wherein executing the operation further comprises:

shifting the ALU result.

22. The method of claim 20 wherein executing the operation comprises:

selecting a field result, on a bit-by-bit basis according to the operand field, from the first and second ALU operands and the ALU result.

23. The method of claim 21 wherein executing the operation comprises:

selecting a field result, on a bit-by-bit basis according to the operand field, from the first and second ALU operands, the ALU result, and the shifted ALU result.

24. The method of claim 23 wherein generating the ALU result comprises:

generating the field result using N single bit ALUs connected in cascade, the field result including a single bit ALU result.

25. The method of claim 24 wherein the field result includes at least a condition code representing a condition of the field result.

26. The method of claim 24 wherein generating the field result comprises:

performing an add/ subtraction on the first and second ALU operands; and

generating a carry output.

27. The method of claim 26 wherein generating the field result further comprises:

generating a zero condition code using a carry from a less significant section;

generating a sign bit for the field result using current and next operand fields; and

generating an overflow bit for the field result using the next operand field.

28. The method of claim 17 further comprising:

generating at least one of the begin and end position specifiers using a field specifier selector.

29. The method of claim 28 wherein generating at least one of the begin and end position specifiers comprises generating the at least one of the begin and end position specifiers from at least one of an instruction specifying the operation, a general-purpose register, a special-purpose register, and a configuration register.

30. The method of claim 16 wherein the operand field is one of a contiguous field and a non-contiguous field.

31. A processor core comprising:

a register file having a plurality of registers;

a condition code register to store condition codes resulted from an operation; and

a field processing unit coupled to the register file and the condition code register to perform an operation, the field processing unit comprising:

a mask generator to generate a mask field for an operand having a word length, the mask field defining an operand field within the operand to be operated by the operation, the operand field having a field length, and

an execution unit coupled to the mask generator to execute the operation on the operand field, leaving portion outside the operand field unchanged.

32. The processor core of claim 31 wherein the mask generator comprises:

a first decoder to decode a begin position specifier into a begin bit pattern;

a second decoder to decode an end position specifier into an end bit pattern; and

a logic circuit coupled to the first and second decoders to combine the begin and end bit patterns into the mask field having the field length limited by the begin and end positions.

33. The processor core of claim 31 wherein the execution unit comprises:

a field arithmetic logic unit (ALU) to generate an ALU result using one of an arithmetic and logic operations on the operand field of at least one of first and second ALU operands.

34. The processor core of claim 33 wherein the execution unit further comprises:

an operand selector to select a selector operand from a source operand and an immediate operand, the source operand being from a register file.

35. The processor core of claim 34 wherein the execution unit further comprises:

a first barrel shifter coupled to the operand selector to shift the selector operand to generate the first ALU operands according to one of the begin and end positions.

36. The processor core of claim 34 wherein the execution unit further comprises:

a second barrel shifter coupled to the operand selector to shift the ALU result.

37. The processor core of claim 35 wherein the execution unit further comprises:

a context selector coupled to the field ALU to select a field result, on a bit-by-bit basis according to the operand field, from at least one of the first and second ALU operands and the ALU result.

38. The processor core of claim 36 wherein the execution unit further comprises:

a context selector coupled to the field ALU to select a field result, on a bit-by-bit basis according to the operand field, from at least one of the first and second ALU operands, the ALU result, and the shifted ALU result.

39. The processor core of claim 36 wherein the field ALU comprises:

N single bit ALUs connected in cascade to generate the field result, the field result including a single bit ALU result.

40. The processor core of claim 39 wherein the field result includes at least a condition code representing a condition of the field result.

41. The processor core of claim 39 wherein the single bit ALU comprises:

an adder/ subtractor to perform an add/ subtraction on the first and second ALU operands and generate a carry output.

42. The processor core of claim 41 wherein the single bit ALU further comprising:

a zero section to generate a zero condition code using a carry from a less significant section;

a negative section to generate a out sign bit for the field result using current and next operand fields; and

an overflow section to generate an overflow bit for the field result using the next operand field.

43. The processor core of claim 32 wherein the field processing unit further comprises:

a field specifier selector coupled to the mask generator to generate at least one of the begin and end position specifiers.

44. The processor core of claim 43 wherein the field specifier selector generates the at least one of the begin and end position specifiers from at least one of an instruction specifying the operation, a general-purpose register, a special-purpose register, and a configuration register.

45. The processor core of claim 31 wherein the operand field is one of a contiguous field and a non-contiguous field.